学　　号：2023200815



北京化工大学

深度学习课程报告

题　目　基于深度学习的网络入侵检测系统

专　　业　计算机科学与技术

学　　生　李泽贤

班　　级　信研 2305

2023 年 12 月 31 日

第 1 章 绪论

随着计算机的发展，新的发展机遇层出不穷的同时，千奇百怪的网络攻击也给互联网的发展带来消极的影响。DoS 攻击、网络钓鱼攻击、特洛伊木马、僵尸网络等网络攻击手段的整体数量保持持续上涨。面对日益复杂且多样的网络攻击，光靠防火墙、杀毒软件等静态防御手段是不够的。

入侵检测系统(Intrusion Detection System, IDS) 可以通过数据来源和检测技术划分。如图 1-1 所示，早期的 IDS 大多都是用基于误用的技术来检测网络攻击，但因为基于误用的入侵检测系统非常依赖已有的签名知识库，对于未知攻击的识别较差。

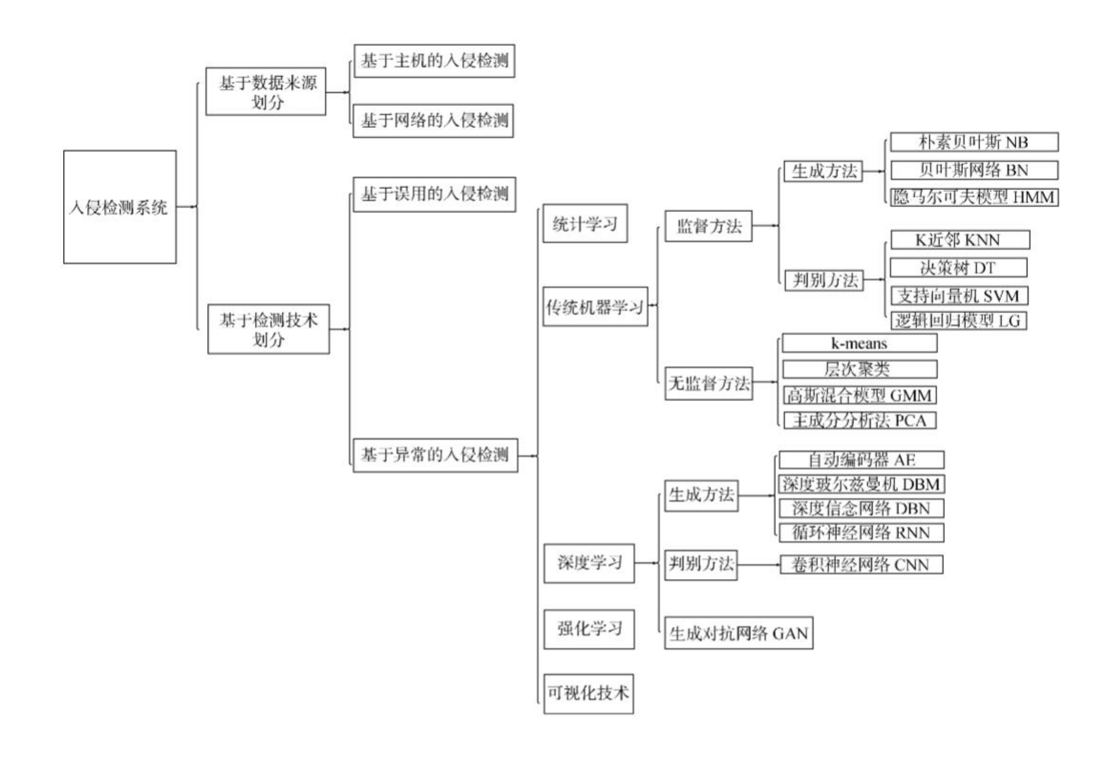


图 1-1 入侵检测系统分类

凭借着深度学习的支持，网络入侵检测的准确率和效率得到提高。为了进一步减少误报率，深度学习网络模型需要在训练中不断改进参数，在预测时防止过拟合，这代表着我们在训练网络模型时的参数量以及结构复杂程度要远超从前。这种情况在一定程度对我们选择的算法和模型有着更高的要求。因此，在保证预测精度的基础上，如何简化网络结构，提高运算效率，减少计算量和功耗成为了工程落地需解决的问题。

第 2 章 相关理论基础

2.1 入侵检测

入侵检测（IDS）不同于防火墙，是一种可以主动监测整个网络，实时监测网络中的入侵行为。如果将防火墙比作商场大门的锁，那么入侵检测就是商场的监控系统。门前大锁可以使部分小偷无法进入商场，而监控系统则可以发现情况并发出警告。入侵检测作为防火墙的重要补充成为构建网络安全防御体系的重要环节，起到了克服传统防御机制的限制。入侵检测在网络中的位置通常是交换机和集线器之间。入侵检测的工作流程如图 2-1。第一步是主机进行信息收集，收集的主要内容是当前系统或用户的相关行为，如：系统日志、应用日志、网络数据、审计记录等信息。其次是入侵分析，将所收集的数据与数据库中的知识库进行对比分析。知识库来自于网络的历史行为或下载的流量特征。分析的方法分为异常检测和误用检测。通过训练好的检测模型，对上一步获取的相关信息进行判断。如果信息与模型中已知的正常行为不匹配，则判断为攻击行为。数据分析是入侵检测中最核心的部分。最后是告警响应，当系统将某一行为判断为攻击，先将此行为的信息保存在数据库中，后将采取向系统发出警告或人工干预来应对攻击。

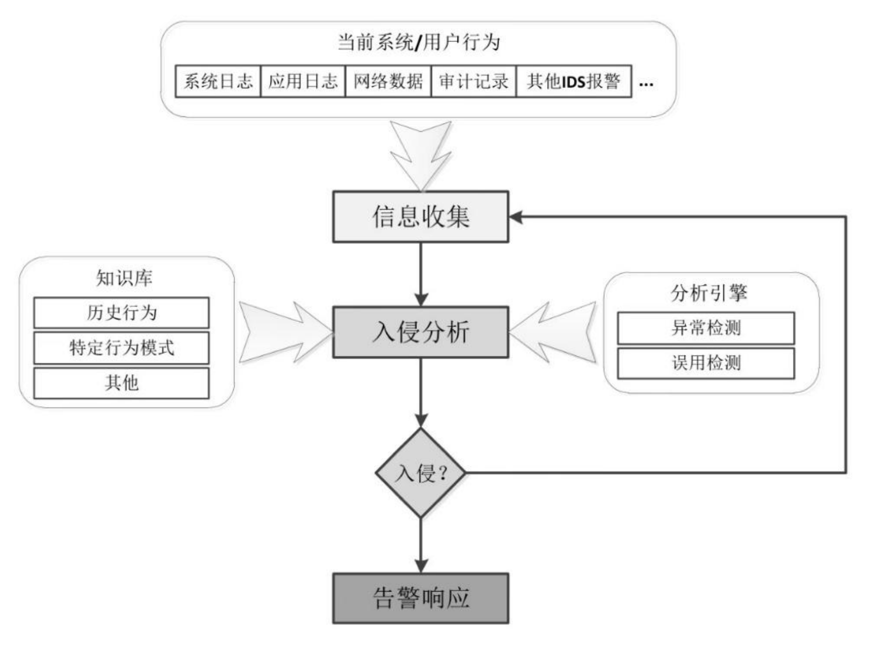


图 2-1 入侵检测工作流程

2.2 基于深度学习的网络入侵检测算法

传统网络入侵检测具有精确度差，效率低，时间长，误报率高等问题，对于大型数据集的检测结果较差。而基于深度学习的网络入侵检测算法的出现，很好的改善了这些问题。

深度学习是一种训练深层神经网络的机器学习算法，其本质是通过构建多层网络的模型并且期待多层的高层次特征来表示数据的抽象语义信息以及通过大量的训练数据来进行特征学习，从而最终提高分类或预测的准确性。深度学习和传统的机器学习在数据的预处理方面比较相似，都对数据进行清洗、归一化、去噪、降维等操作。但传统机器学习相比于深度学习，其对数据的特征提取主要依赖人工提取，面对简单任务时效率高且可解释性强，但不通用。而深度学习是机器提取特征，分类表现相对较好但可解释性差。

2.2.1 K-最近邻算法

KNN 算法全称是 K-最近邻算法(K-NearestNeighbor)，是基于实例的学习，是懒惰学习的代表。KNN 算法具有广泛的应用场景，在垃圾邮件识别、图像内容识别、文本分析都有大量的应用。算法的核心思想非，即给定一个训练数据集，将输入的数据在训练数据集中找到与该数据最邻近的 K 个实例。根据 K 个实例中的多数类来对新实例进行分类。通过取它们的加权平均值来解决回归问题。如图 2-2 所示。

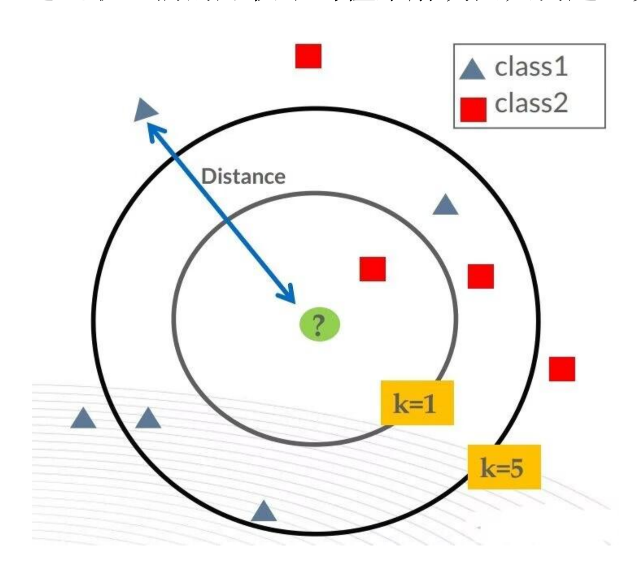


图 2-2 KNN 算法分类

圆圈的大小表示 k 的大小。圆圈里哪个类别的数量最多，那么预测值就是哪个类别。当 k=1 时，分类到红色方块；当 k=5 时，由于圈中灰色三角形较多，故分类到灰色三角形。我们总结发现最终的预测值与两个因素有关：k 值的大小和距离判断的方法。

常用的欧式距离：

是我们学习最直观的两点之间或多点之间的距离表示法，如图 2-3 所示，它可以简单理解为两个点之间连线的长度。如 A(x1,y1)和 B(x2,y2)之间的欧式距离为：

$$d(A,B)=\sqrt{(x_1-x_2)^2-(y_1-y_2)^2} \quad (2\text{-}1)$$

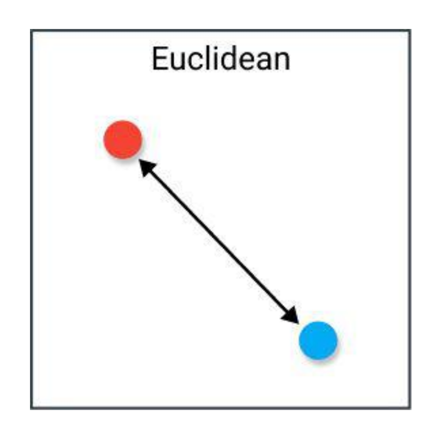


图 2-3 欧氏距离

2.2.2 BP 神经网络

BP(Back-propagation，反向传播)神经网络是最传统最简单的神经网络，因为其参数训练思想使其成为应用最广泛的神经网络。所谓误差反向传播，就是对损失函数和权重值利用链式法则求导（每个权值参数都可以看做是损失函数的参数）。因为梯度的反方向就是函数值下降最快的方向，我们可以利用梯度下降来优化损失函数。通过以上两点对权重值进行更新直至误差函数最小，模型训练结束。

以下图 2-9 所示，在一个有两个输入和一个输出的三层神经网络中，每个神经元都是执行两次操作，首先将权重系数与输入信号进行乘积，其次将乘积放入神经元激活函数。

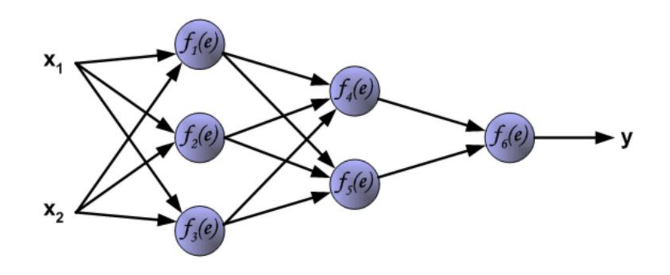


图 2-9 BP 神经网络实例

为了训练神经网络模型，我们需要准备训练集，其中包括输入信号(x1,x2)和相

应目标（期望输出）z。网络训练是一个迭代过程，每个迭代中会使用来自训练数据集的新数据来修改节点的权重系数。这种修改是通过算法如下图 2-10 所示来完成的，每个步骤都需要两个来自训练集的输入信号。在网络中传播信号时，我们可以确定每个神经元在每个网络层中的输出信号值，其中符号 w(xm)n 表示输入层中网络输入 xm 和神经元 n 之间的连接权重，符号 yn 表示神经元 n 的输出信号。

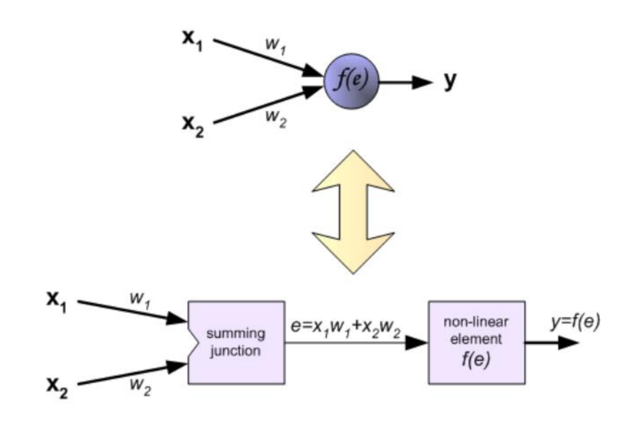


图 2-10 反向传播

调整网络参数来使网络输出的结果尽可能的靠近真实值的方式至关重要。这里先引入一个概念——损失函数。所谓损失(loss)，就是网络的输出值 y_{out} 和真实值 y 之间的误差$|y_{out}-y|$。而计算损失的函数即为损失函数。常用的损失函数有均方差函数和交叉熵损失。网络模型更优就相当于使损失函数变小，使损失函数变小的方法是梯度下降法。其中梯度是将由不同参数下的损失函数构成的曲线进行求导所得来的，梯度越小代表损失函数下降的越快，由此得到损失函数的最小值。优化网络模型参数是通过调整每个神经元输入的权重系数来实现的，我们可以由公式 2-10 来更新每个神经元的参数（不同输入的权重系数）。公式中的 df(e)/de 表示每个神经元激活函数的导数。

$$w'_{(x1)1} = w_{(x1)1} + \eta \delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta \delta_1 \frac{df_1(e)}{de} x_2 \tag{2-10}$$

所谓的训练网络，就是模型在不断正向传播和反向传播，不断优化模型参数，逼近模型最优性能。理论上不断的训练可以最终达到最优模型，但实际上训练到一定程度时最终的分类效果反而会变差，这也是近几年研究的热点

第 3 章 基于深度学习的网络入侵检测算法的训练集和框架

3.1 网络训练的数据集

本次实验使用的是 KDD CUP 99 数据集。此数据集是入侵检测中最常用的，来自于 1998 年的美国国防部高级规划署(DARRA)入侵检测评估项目，后 Wenke Lee 等人对原始数据进行特征提取，其数据来自于美国空军局域网，进行了 9 周网络流量收集，其中 7 周流量为训练集，剩余两周为测试集。大约包含 700 万流量。在测试集中还包含了一些训练时不包含的攻击类型，用来测试入侵检测模型对未知攻击的识别能力。

训练集中将网络流量分为正常和异常，在流量标签中说明类型。如下图 3-1 所示，异常流量分为了 4 大类，4 大类中总共有 39 种攻击类型，但只有 22 个攻击类型在训练集，剩下的 17 种类型在测试集。对未知网络攻击的识别能力是评测网络入侵检测系统的重要指标。

标识类型	含义	具体分类标识
Normal	正常记录	normal
DOS	拒绝服务攻击	back、land、neptune、pod、smurf、teardrop
Probing	监视和其他探测活动	ipsweep、nmap、portsweep、satan
R2L	来自远程机器的非法访问	ftp_write、guess_passwd、imap、multihop、phf、spy、warezclient、warezmaster
U2R	普通用户对本地超级用户特权的非法访问	buffer_overflow、loadmodule、perl、rootkit

图 3-1 KDD CUP 99 训练的标识类型

KDD CUP 99 数据集中的每个流量记录包含了 41 个固定的特征属性和 1 个类标识，其中 9 个特征属性为离散(symbolic)型，其它均为连续(continuous)型。

第 4 章 实验与结果分析

本次毕业设计实验在 PC 端个平台实现。本章将介绍 PC 端软硬件的配置，并对所使用的数据集 KDD CUP 99 的数据预处理环节进行介绍。最后对网络入侵检测系统的训练和预测结果进行展示与分析。

4.1 数据集预处理

本次毕业设计使用 KDD CUP 99 数据集中包含有符号型的数据属性，在神经网络模型中，无法处理非数值型的特征。为了更好的进行实验，我们需要对数据进行预处理，数据集的预处理一般分为三个步骤：1、将数据集中字符型数据转换为数值型数据。2、数值标准化。3、数值归一化。

4.1.1 字符型数据转换为数值型数据

在 KDD CUP 99 数据集的 41 个特征中有 3 个符号型特征，这 3 个符号型特征分别是协议类型特征(protocol_type)、目标网络服务特征(service)、连接正常与否的状态特征(flag)。对于这 3 个符号型特征，我们可以使用属性映射的方法将字符型数据转换为数值型数据[23]。

协议类型特征就是网络流量的传输层协议，有 3 种类型 TCP(传输控制协议 Transmission Control Protocol)、UDP(用户数据报协议 User Datagram Protocol)、ICMP(Internet 控制报文协议 Internet Control Message Protocol)。将其编码为 0-2，用二进制编码为 TCP-00、UDP-01、ICMP-10。目标网络服务特征有 70 种，如 http、ftp、IRC、link、login、mtp 等等，对其编码 0-69。连接正常与否的状态特征(flag)有 11 种，如 OTH、REJ、RSTR 等等，对其编码 0-10。最后将攻击类型进行编码，正常流量为 0，Dos-1、Probe-2、R2L-3、U2R-4。

4.2 实验流程

本次实验首先进行的是划分训练集和测试集，将 KDD CUP 99 数据集的 20%划分为测试集，剩下的为训练集。后进行数据的预处理，通过独热编码将字符型数据转换为数值型数据，再进行 Z-Score 标准化和 Max-Min 归一化。接下来确定训练的目标，如果是二分类就判断数据的标签，若标签为异常则置 1。在 K 等于 3、5、7、9、11、13、15 情况下使用 KNN 算法并放入训练集及对应类别。再对测试集进行预

测，计算输入点与邻近点的欧式距离并进行排序，取前 k 个点。最终这几点中的类别最多的则是输入点的类别。将预测的结合和标签进行对比计算正确率。最后利用函数生成不同 k 值下的正确率。工作的流程图如图 4-1 所示：

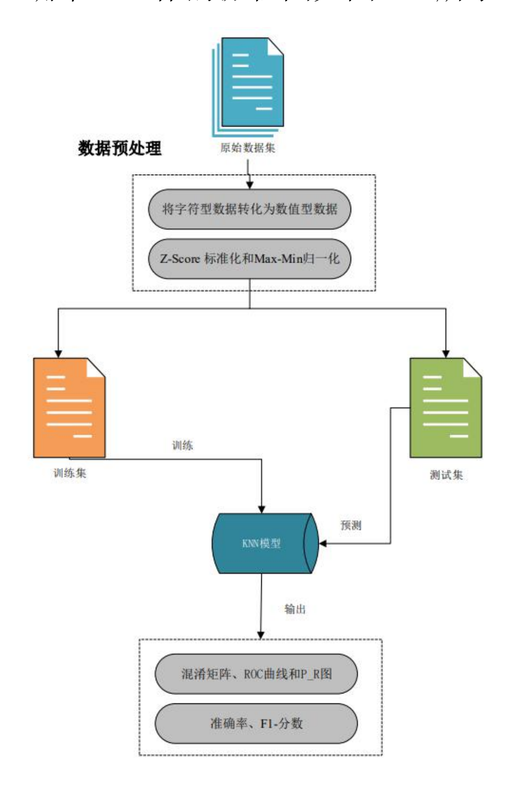


图 4-1 基于 KNN 算法下入侵检测的流程图

找到正确率最高对应的 k 值，后套用 BP 神经网络、朴素高斯贝叶斯和决策树算法对测试集进行预测。生成 ROC 曲线、P_R 曲线和混淆矩阵将对比其它模型。

4.3 实验结果与分析

本次实验使用 KNN 算法进行训练以及预测，后将高斯贝叶斯、BP 神经网络和决策树一同进行并输出分类结果进行对比。对于数据集 KDD CUP 99，分别展示二分类（正常和异常）和五分类（正常和 4 大类攻击）下的结果与分析。由于原始数据集过多且本主机性能不佳，为加快实验进度，将交叉验证法的折数均设为 5，且只抽取原始数据集的 20%约 8w 条数据包作为训练集和测试集，其中测试集占 20%。

（1）二分类

为防止数据不足导致实验精度下降和估计模型在实际数据应用中的准确度，本实验使用交叉验证法。

在 KNN 算法中，我将 K 参数值选定在 3-15 之间，每次增加 2(共 3、5、7、9、

11、13、15 这几种情况)，分别计算对应 k 值下的平均准确率，如图 4-2 所示：

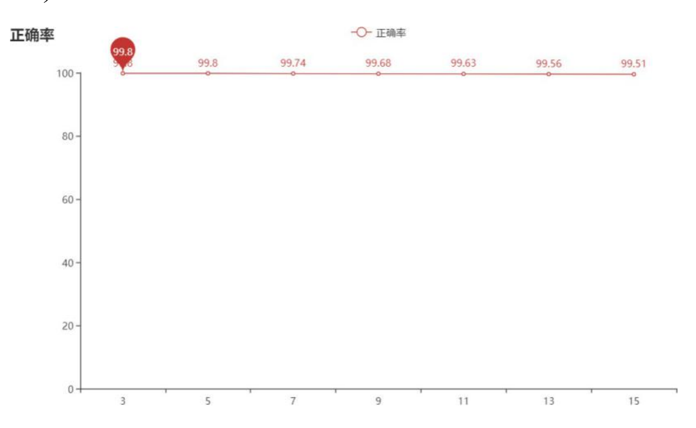


图 4-2 不同 k 值下的入侵检测率

所有 k 值下的准确率都在 99%以上，都符合实验前的预期。其中 K=3 时平均准确率最高，最终选择 K=3 作为二分类下的 KNN 算法的 K 值。

后将 KNN 算法与其它算法进行比较，首先选择决策树算法，将决策树参数 max_depth 的范围划定为 10 到 30，各值下的正确率如图 4-3 所示：

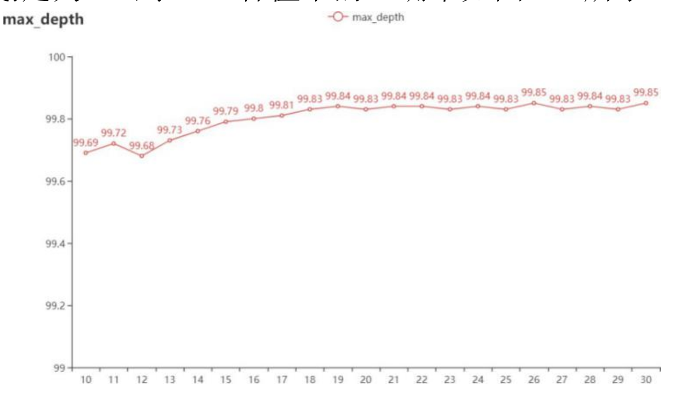


图 4-3 10-30max_depth 值下的正确率

可以看到等于 26 和 30 的时候最高，这里取值为 26，在此设定下，在对测试范围为 2 到 20 的进行正确率的检测，如图 4-4 所示：

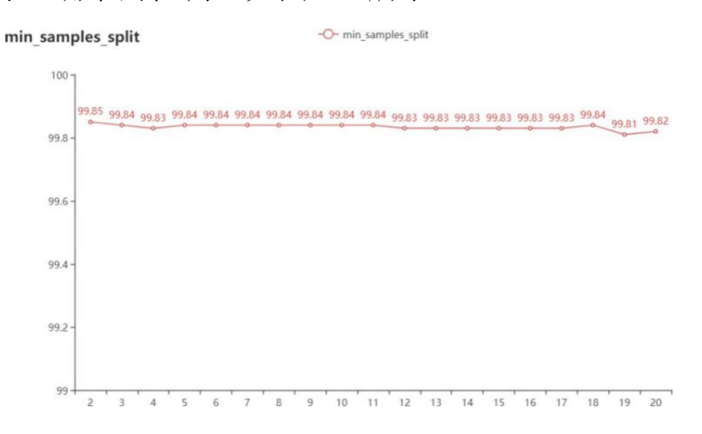


图 4-4 2-20max_depth 值下的正确率

最终 min_samples_split 设定为 **2**。

性能评价：

首先先看图 4-5 各模型下的混淆矩阵：

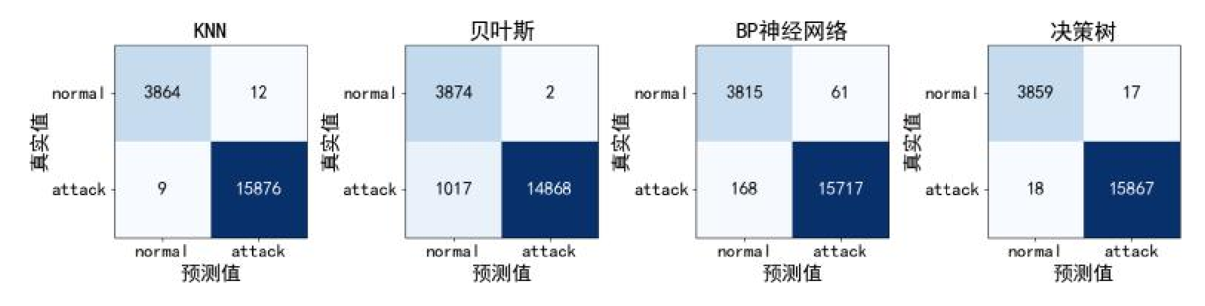


图 4-5 各模型下的混淆矩阵

图 4-6 是各模型的准确率、精确率、召回率和 F1-Score（这里我将结果都扩大了 100 倍）：

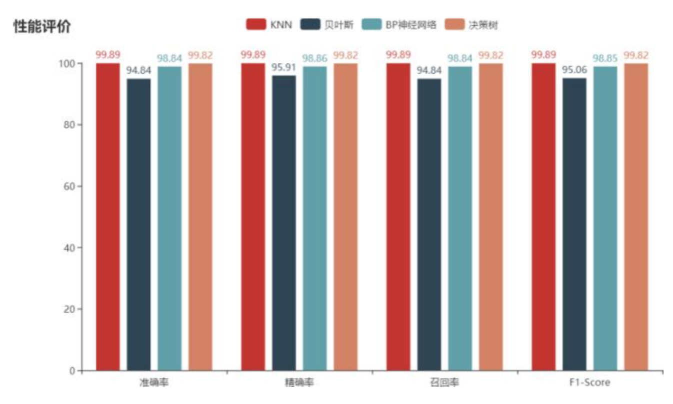


图 4-6 各模型下的准确率、精确率、召回率和 F1-Score

最后是 ROC 曲线和 P_R 曲线：

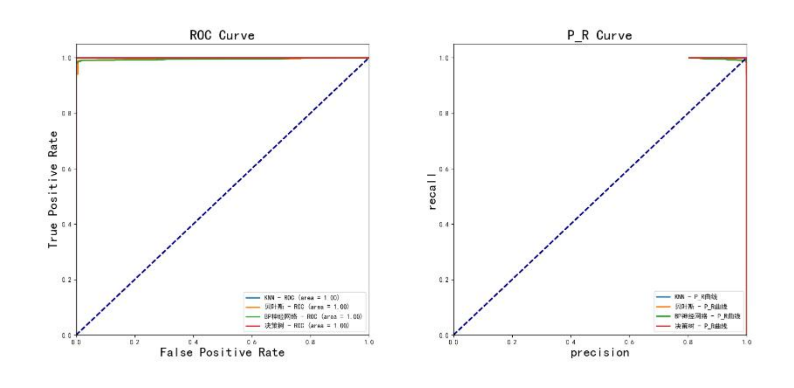


图 4-7 各模型下的 ROC 曲线和 P_R 曲线

ROC 曲线中，每一个模型的 AUC 面积保留两位小数点后均为 1，四个模型之间的差距不大，正确率等指标在 94% 之上，分类结果均较好。在 P_R 曲线中也难以比较各模型之间的优劣。但观察混淆矩阵，我们发现这几个模型中 KNN 表现得最好。但这四个模型都存在着将正常数据包识别为异常数据包的情况。

（2）五分类

在五分类的 KNN 算法下，我将 K 参数值选定与二分类一致，分别计算对应 K 值下的平均准确率，如图 4-8 所示：

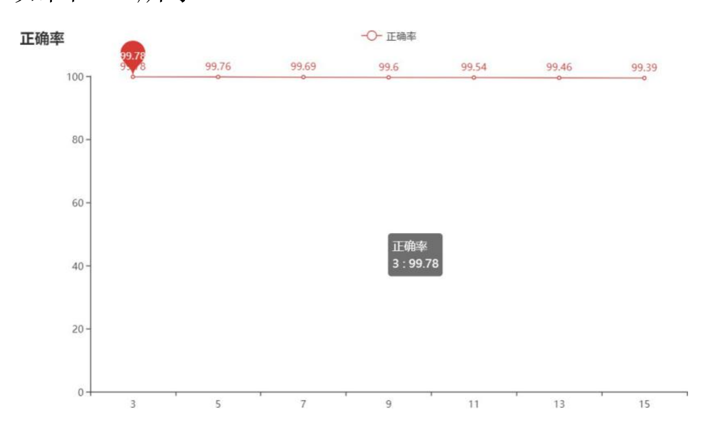


图 4-8 不同 k 值下的入侵检测率

由结果所示，五分类下的 KNN 算法的 K 值的设置同二分类下的一致为 3。

同样选定决策树算法与之比较，决策树 max_depth 参数取值范围 10 到 30，正确率如图 4-9：

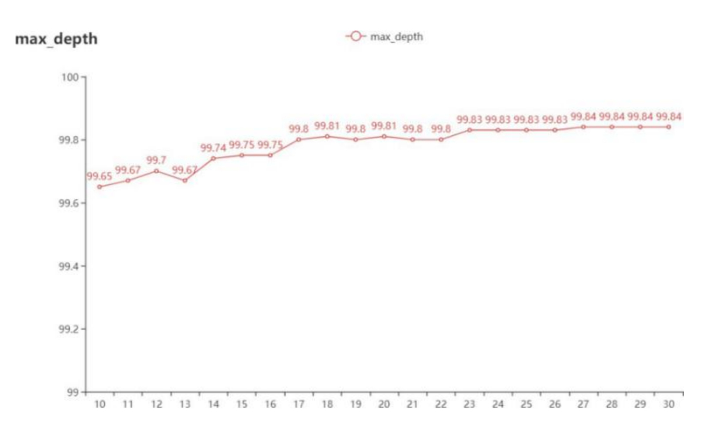


图 4-9 10-30max_depth 值下的正确率

在参数到 27 的时候，正确率基本就保持在 99.84，因此 max_depth 设定为 **27**。在此基础上再对 min_sampls_split 进行测试，范围是 2 到 20，正确率如图 4-10：

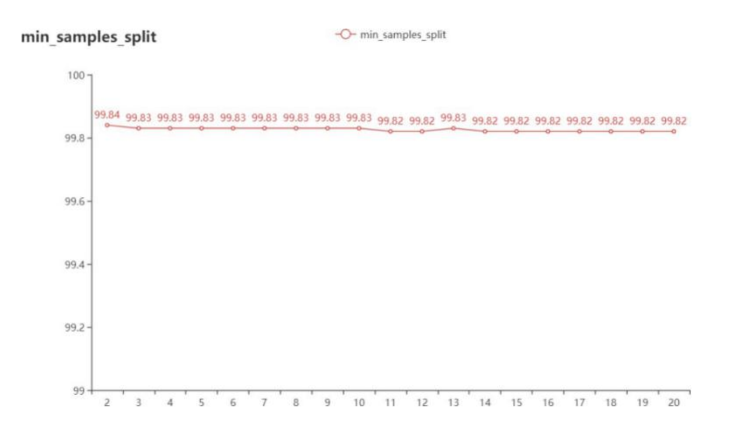


图 4-10 2-20max_depth 值下的正确率

随着该参数的增大，正确率呈下降趋势，因此该参数最终设定为 **2**。

性能评价

混淆矩阵的对比如图 4-11：

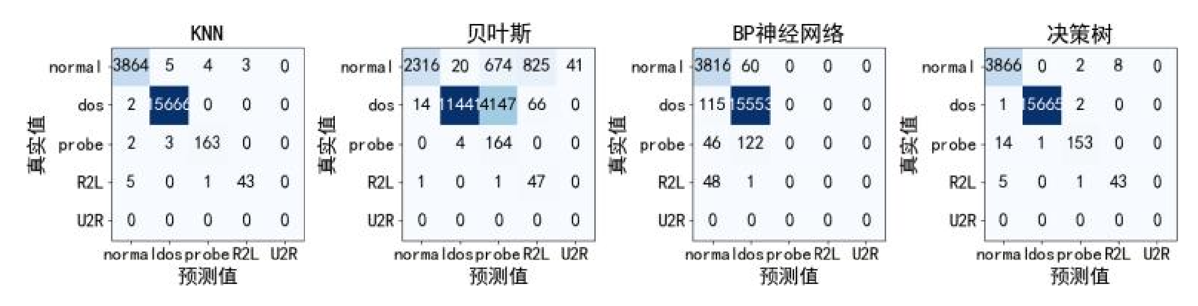


图 4-11 各模型下的混淆矩阵

各模型的准确率、精确率、召回率和 F1-Score 比较如图 4-12：

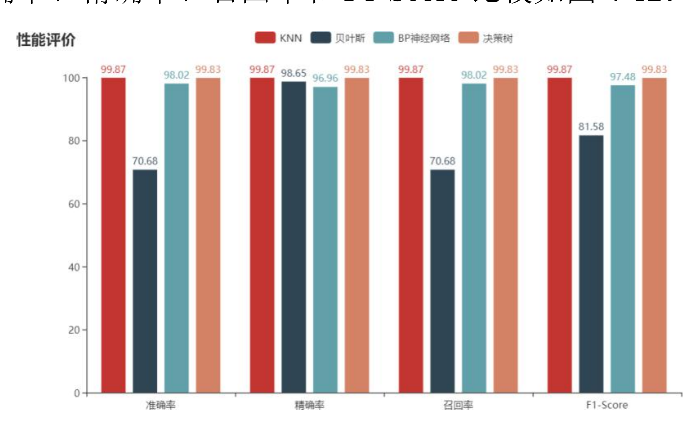
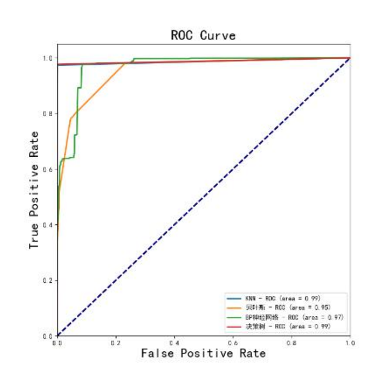


图 4-12 各模型下的准确率、精确率、召回率和 F1-Score

ROC 曲线和 P_R 曲线的对比如图 4-13：



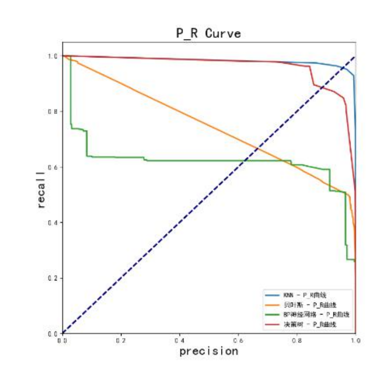


图 4-13 各模型下的 ROC 曲线和 P_R 曲线

将分类从二分类改为五分类后，各模型的预测表现明显不同。尽管四个模型的 ROC 曲线存在交叉，但贝叶斯的 AUC 面积最小，而 KNN 和决策树的面积最大。在 P_R 曲线中，BP 神经网络的对角线交点最低，而 KNN 的对角线交点最高。从总体来看，贝叶斯在准确率和混淆矩阵中表现的效果最差，KNN、神经网络和决策树的正确率等性能相似，但混淆矩阵可以具体反应出他们的差距。相对而言，KNN 和决策树的分类更具泛化性。BP 神经网络的性能高是因为数据集不平衡（DOS 攻击的数据包在 KDD CUP 99 数据集中占比较大），而对其他攻击和正常数据包的识别率很低。在样本不平衡的影响下，对 DOS 攻击的异常数据包最多，对其提取的特征也越多，而其他攻击的数据包非常少提取的特征也少，最终导致训练阶段的收敛和测试的泛化性降低。样本不平衡性对其他的模型也产生了影响，但高斯贝叶斯在少数类判断上会更为成功一些，但整体效果更差。综合来看，KNN 的效果最好。在二分类中，所有异常数据包都合成了一个类别，所以样本不平衡性没有那么明显，四个模型表现也更好。

结 论

随着深度学习的快速发展和迭代，越来越多的网络模型如雨后春笋般出现。为了追求更快速的分类，各个领域都开始使用深度学习框架。而在信息安全越来越重要的今天，传统的网络入侵检测已经无法做到实时精准的检测，基于深度学习的网络入侵检测应运而生。本次实验基于 Sklearn 框架，采用 KDD CUP 99 数据集，训练基于 KNN 算法二分类和多五分类的网络模型，生成混淆矩阵、ROC 曲线和 P_R 曲线，通过对比朴素高斯贝叶斯、BP 神经网络、决策树等算法和不同 k 值下 KNN 算法的精确度来，比较各算法的优劣和确定 k 的值。通过 Pyecharts 包实现数据可视化，生成准确率的图。本次实验的预测效果整体良好，符合之前预期，相比于网上其它算法有着不小的优势。